대 한 민 국 특 허 청
KOREAN INTELLECTUAL PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto is a true copy from the records of the Korean Intellectual Property Office.

출 원 번 호 : 10-2002-0061179
Application Number PATENT-2002-0061179

출 원 년 월 일 : 2002년 10월 08일
Date of Application OCT 08, 2002

출 원 인 : 삼성전자 주식회사
Applicant(s) SAMSUNG ELECTRONICS CO., LTD.

2002 년 11 월 09 일

특 허 청
COMMISSIONER

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2002.10.08
【국제특허분류】	G09C
【발명의 명칭】	무선 통신 시스템에서 암호화 장치 및 방법
【발명의 영문명칭】	APPARATUS AND METHOD FOR CIPHERING IN MOBILE COMMUNICATION SYSTEM
【출원인】	
【명칭】	삼성전자주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	임종수
【성명의 영문표기】	LIM,Jong-Su
【주민등록번호】	700114-1046519
【우편번호】	442-380
【주소】	경기도 수원시 팔달구 원천동 35 원천 주공 아파트 1018-407
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사 를 청구합니다. 대리인 이건주 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	18 면 18,000 원

【우선권주장료】	0 건	0 원
【심사청구료】	10 항	429,000 원
【합계】	476,000 원	

【요약서】

【요약】

본 발명은 무선 통신 시스템에서 암호화 알고리즘에 관한 것으로, 입력신호와 동일한 비트의 암호화 신호를 출력하는 알고리즘을 제공하는 것으로, 입력 신호를 병렬 연산하여 암호화하는 장치 및 방법을 제공하는 것이다. 또한, 내부 지연에 따른 지연신호와 지연이 발생하지 않은 신호의 출력 동기를 맞추어 최종적으로 정확한 신호를 출력하는 암호화 장치 및 방법을 제공하는 것이다. 따라서, 입력된 신호들을 암호화 연산함에 있어서 연산 처리 속도의 효율성을 제공하며, 내부 지연에 따른 지연 동기를 맞추기 위해 사용되는 소자의 수를 축소시키는 효과가 있다. 따라서, 적은 소자 사용으로 인한 논리 연산시 암호화 시스템의 안정성과 하드웨어 구성시 비용을 절감하는 효과가 있다.

【대표도】

도 5

【색인어】

KASUMI알고리즘, UMTS(Universal Mobile Telecommunication System), SBox7, SBox9,

【명세서】

【발명의 명칭】

무선 통신 시스템에서 암호화 장치 및 방법{APPARATUS AND METHOD FOR CIPHERING IN MOBILE COMMUNICATION SYSTEM}

【도면의 간단한 설명】

도 1은 종래 기술에서 제안하고 있는 KUSUMI 알고리즘을 도시한 도면.

도 2a는 종래 기술에 따른 FOi부 상세 구성의 일 예를 도시한 도면.

도 2b는 종래 기술에 따른 FOi'부 상세 구성의 다른 예를 도시한 도면.

도 3은 종래 기술에 따른 FI서브암호화부의 상세 구성을 도시한 도면.

도 4는 본 발명에서 제안하고 있는 KASUMI 알고리즘을 도시한 도면.

도 5는 본 발명에 따른 슬림FOi부 상세 구조를 도시한 도면.

도 6은 본 발명에 따라 FI서브암호화부의 상세 구성을 도시한 도면.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

<8> 본 발명은 무선 통신 시스템에 관한 것으로, 특히 암호화 알고리즘과 무결성 알고리즘을 구현하는 암호화 장치 및 방법에 관한 것이다.

<9> 아날로그 제 1세대 시스템으로부터 디지털 제 2세대 시스템으로 바뀌면서, 보다 진보된 암호화 방법들이 사용되고 있다. 고도의 정보화 사회에서 요구되는 음성신호와 영상신호등의 멀티 미디어 서비스를 제공하는 제 3세대 시스템에서는 사용자에 관한 정보는 물론, 음성신호 및 멀티 미디어 서비스 제공에 따른 신호들의 기밀성을 개선하기 위한 암호화 알고리즘에 관한 중요성이 증가하고 있다. 또한, 이동 단말기와 네트워크간의 제어 신호를 인증하기 위한 무결성 알고리즘의 중요성이 증대되고 있다. 따라서, GSM(Global System For Mobile Communication)핵심망에 기반한 제 3 세대 시스템(UMTS: Universal Mobile Telecommunication System)의 3GPP(The 3 Generation Project Partnership)에서는 전 세계적으로 적용 가능한 표준화된 암호화 알고리즘(f8)과 무결성 알고리즘(f9)으로 KASUMI 알고리즘을 정의하고 있다.

<10> 도 1은 종래 기술에서 제안하고 있는 KASUMI 알고리즘을 도시한 도면이다.

<11> 상기 도 1을 참조하면, KASUMI 알고리즘은 8-라운드 페이스탈(Feistel) 구조를 갖는 블록 암호 시스템으로, 64비트 평문신호(plaintext)를 입력하여 8번의 암호화 단계를 거쳐 64비트의 암호화신호(ciphertext)로 출력한다. 상기 64비트의 입력신호는 분할되어 32비트의 L0 신호와 32비트의 R0 신호로 전송된다. 즉, 상기 32비트의 L0신호와 상기 32비트의 R0 신호는 복수 개의 FLi부들($1 \le i \le 8$)(110,120,130,140,150,160,170,180)와 복수 개의 FOi부들($1 \le i \le 8$) (210,220,230,240,250,260,270,280)에서 상기 각각의 암호화부들에 해당하는 각각의 암호화 키 KLi($1 \le i \le 8$), KOi($1 \le i \le 8$), KIi($1 \le i \le 8$)에 의해 암호화되어 64비트의 암호화 신호를 출력한다.

<12> 다음에서 상기 암호화 단계를 좀더 구체적으로 설명하고자 한다. 제 1 암호화 단계를 살펴보면, 입력된 32비트 L0신호는 첫 번째 FL1부(110)에서 첫 번째 암호화키 KL1과

연산하여 암호화 신호 LO1를 출력한다. 첫 번째 FO1부(210)에서 상기 암호화 신호 LO1은 각각의 첫 번째 암호화키 KO1와 첫 번째 암호화 키 KI1에 의해 암호화되어 32비트의 LO2 신호를 출력한다. 상기 첫 번째 FO1부(210)에서 상기 LO2신호는 상기 32비트의 RO신호와 배타적 논리합 연산하여 암호화된 64비트 신호를 출력한다. 따라서, 상기 KASUMI 알고리즘은 64비트 신호를 입력하여 상기 제 1 암호화 단계와 같이 8번 암호화 과정을 반복하여 최종적으로 64비트의 암호화 신호를 출력한다.

<13> 도 2a는 상기 도 1에서 FOi부 상세 구성의 일 예를 도시한 도면이다.

<14> 상기 2a를 참조하면, FOi부는 i번째 FO부를 말하는 것으로, 상기 FOi부는 복수개의 FIi,j서브암호화부들(1≤i≤3이고, 1≤j≤3이다)로 구성되어 3단계의 암호화 과정을 이루어진다. 여기서는 1번째 FO부를 예로 들어 설명하고자 한다. 32비트의 입력 신호는 각각 16비트의 LO신호와 16비트의 RO신호로 분할되어 전송된다. 상기 3단계 중 단계 1을 설명하면, 16비트의 LO신호는 16비트의 서브 암호화키 KO1,1과 배타적 논리합 연산하여 L1신호를 출력한다. 상기 L1신호는 FI1,1서브암호화부(201)에 의해 16비트의 서브 암호화키 KI1,1와 암호화되어 지연된 L1D신호를 출력한다. 반면에, 16비트의 RO=R1신호는 지연기D1(10)을 통과하면서 지연된 R1D신호를 출력한다. 즉, 상기 지연기D1(10)은 상기 R1D신호를 임의로 지연시킴으로써 상기 L1D신호와의 출력 동기를 맞추는 역할을 한다. 상기 3단계 중 단계 2를 설명하면, 16비트의 R1D신호는 16비트의 서브 암호화키 KO1,2과 배타적 논리합 연산하여 L2신호를 출력한다. 상기 L2신호는 FI1,2서브암호화부(203)에 의해 16비트의 서브 암호화키 KI1,2와 암호화되어 지연된 L2D신호를 출력한다. 반면에, 16비트의 R1D신호는 상기 L1D신호와 배타적 논리합 연산을 하여 R2신호를 출력한다. 상기 R2신호는 지연기D2(20)을 통과하면서 지연된 R2D신호를 출력한다. 즉, 상기 지연기

D2(20)는 상기 R2신호를 임의로 지연시킴으로써 상기 L2D신호와의 출력 동기를 맞추는 역할을 한다. 상기 3단계 중 단계 3를 설명하면, 16비트의 R2D신호는 16비트의 서브 암호화키 KO1,3과 배타적 논리합 연산하여 L3신호를 출력한다. 상기 L3신호는 FI1,3서브암호화부(205)에 의해 16비트의 서브 암호화키 KI1,3와 암호화되어 지연된 L3D신호를 출력한다. 반면에, 16비트의 R2D신호는 상기 L2D신호와 배타적 논리합 연산을 하여 R3신호를 출력한다. 상기 R3신호는 지연기D3(30)을 통과하면서 지연된 R3D신호를 출력한다. 즉, 상기 지연기D3(30)는 상기 R3신호를 임의로 지연시킴으로써 상기 L2D신호와의 출력 동기를 맞추는 역할을 한다. 상기 R3D신호는 상기 L3D 신호와 배타적 논리합 연산하여 R4신호를 출력한다. 따라서, 16비트 R4 신호와 16비트 상기 R3D=L4신호 연산하여 32비트의 암호화 신호(L4?‖R4)를 출력한다.

<15> 즉, 상기 FO1부는 상기 서브암호화부들(201,203,205)로부터 출력되는 지연 신호와의 지연되지 않은 신호들의 동기를 일치시키기 위하여 총 3개의 지연기들(10, 20, 30)을 사용한다.

<16> 도 2b는 상기 도 1에서 FO'부 상세 구성의 다른 예를 도시한 도면이다.

<17> 상기 도 2b를 참조하면, FOi'부는 i번째 FO'부를 말하는 것으로, 상기 FOi'부는 복수개의 FIi',j'서브암호화부들(1≤i'≤3이고, 1≤j'≤3이다)로 구성되어 3단계의 암호화 과정을 이룬다. 32비트의 입력 신호는 각각 16비트의 Lo'신호와 16비트의 Ro'신호로 분할되어 전송된다.

<18> 16비트의 입력 신호 LO'신호는 16비트의 서브 암호화키 KO1,1과 배타적 논리합 연산하여 L1'신호가 된다. 상기 L1'신호는 FI1',1'서브암호화부(211)에 의해 첫 번째 16비트의 서브 암호화키 KI1,1와 암호화되어 지연된 L1D'신호로 출력된다. 16비트의 입력신

호 RO'=R1'신호는 지연기D4(40)를 통과하면서 지연된 R1D'신호가 된다. 상기 L1D'신호와 상기 R1D'신호는 배타적 논리합 연산하여 L2'신호를 출력한다. 동시에 상기 16비트의 RO'=SR1'신호는 16비트의 서브 암호화키 KO1,2과 배타적 논리합 연산하여 R2' 신호가 된다. 상기 R2'신호는 FI1',2'서브암호화부(213)에 의해 16 비트의 서브 암호화키 KI1,2와 암호화되어 R2D'신호를 출력한다. 상기 R2D'신호는 상기 SL2'신호와 배타적 논리합 연산하여 R3'신호를 출력한다. 상기 L2'신호는 16 비트의 서브 암호화키 KO1,3과 배타적 논리합 연산하여 L3'신호를 출력한다. 상기 L3'신호는 FI1',3'서브암호화부(215)에 의해 16 비트의 서브 암호화키 KI1,3와 암호화되어 지연된 L3D'신호를 출력한다. 반면에, R3' 신호는 지연기D5(50)을 통과하면서 지연된 R3D'신호를 출력한다. 상기 R3D'신호와 상기 L3D'신호는 배타적 논리합 연산하여 16비트 L4신호를 출력한다. 따라서, 16비트 L4' 신호와 16비트 상기 R3D'=L4'신호 연산하여 32비트의 암호화 신호(L4‖R4)를 출력한다.

<19> 상기 전술한 바와 같이 개량된 FO1'부는 총 2개의 지연기들(40,50)을 사용하여 지연 신호와의 지연되지 않은 신호들의 동기를 일치시킨다. 즉, 상기 복수 개의 FIi,j서브 암호화부들로부터 출력되는 출력 신호들의 동기를 맞추기 위해 부가적으로 지연기들을 사용하게 되었다. 따라서, 하드웨어 구성함에 있어서 칩 용량이 대형화되는 문제가 발생하였다.

<20> 도 3은 상기 도 2a와 도 2b에서의 FI서브암호화부의 상세 구성을 도시한 도면이다. 여기서는 첫 번째 FI1서브암호화부를 예로 들어 설명하고자 한다.

<21> 상기 도 3을 참조하면, 16비트의 입력신호는 9비트 RLO신호와 7비트 RRO신호로 분할된다. SBox91연산기(이하 'S91'라 칭함)(310)는 9비트의 입력 신호 RLO를 입력하여 하기의 <수학식 1>에 적용함으로써 9비트의 신호들 Y0,Y1,...Y8를 출력한다.

<22>

$$
\begin{aligned}
y0 &= x0x2 \oplus x3 \oplus x2x5 \oplus x5x6 \oplus x0x7 \oplus x1x7 \oplus x2x7 \oplus x4x8 \oplus x5x8 \oplus x7x8 \oplus 1 \\
y1 &= x1 \oplus x0x1 \oplus x2x3 \oplus x0x4 \oplus x1x4 \oplus x0x5 \oplus x3x5 \oplus x6 \oplus x1x7 \oplus x2x7 \oplus x5x8 \oplus 1 \\
y2 &= x1 \oplus x0x3 \oplus x3x4 \oplus x0x5 \oplus x2x6 \oplus x3x6 \oplus x5x6 \oplus x4x7 \oplus x5x7 \oplus x6x7 \oplus x8 \oplus x0x8 \oplus 1 \\
y3 &= x0 \oplus x1x2 \oplus x0x3 \oplus x2x4 \oplus x5 \oplus x0x6 \oplus x1x6 \oplus x4x7 \oplus x0x8 \oplus x1x8 \oplus x7x8 \\
y4 &= x0x1 \oplus x1x3 \oplus x4 \oplus x0x5 \oplus x3x6 \oplus x0x7 \oplus x6x7 \oplus x1x8 \oplus x2x8 \oplus x3x8 \\
y5 &= x2 \oplus x1x4 \oplus x4x5 \oplus x0x6 \oplus x1x6 \oplus x3x7 \oplus x4x7 \oplus x6x7 \oplus x5x8 \oplus x6x8 \oplus x7x8 \oplus 1 \\
y6 &= x0 \oplus x2x3 \oplus x1x5 \oplus x2x5 \oplus x4x5 \oplus x3x6 \oplus x4x6 \oplus x5x6 \oplus x7 \oplus x1x8 \oplus x3x8 \oplus x5x8 \oplus x7x8 \\
y7 &= x0x1 \oplus x0x2 \oplus x1x2 \oplus x3 \oplus x0x3 \oplus x2x3 \oplus x4x5 \oplus x2x6 \oplus x3x6 \oplus x2x7 \oplus x5x7 \oplus x8 \oplus 1 \\
y8 &= x0x1 \oplus x2 \oplus x1x2 \oplus x3x4 \oplus x1x5 \oplus x2x5 \oplus x1x6 \oplus x4x6 \oplus x7 \oplus x2x8 \oplus x3x8
\end{aligned}
$$

【수학식 1】

<23> ZE1부(320)는 7비트의 RR0신호를 입력받아 최상위비트(Most Significant Bit,이하 'MSB'라 칭함)에 2개의 ZERO(0)비트를 추가하여 9비트의 신호를 출력한다. 상기 S91연산기(310)의 9비트 출력신호와 상기 ZE1부(320)의 9비트 출력신호는 배타적 논리합 연산되어 9비트의 RL1신호로 출력한다. 상기 RL1신호는 9비트의 서브 암호화키 KI1,1,2와 배타적 논리합 연산하여 9비트의 RL2신호를 출력한다.

<24> 반면에, TR1부(330)는 상기 9비트 RL1신호의 MSB 비트 중 ZERO(0)비트에 위치한 2개의 신호들을 제거하여 7비트의 신호를 출력한다. SBox71연산기(이하 'S71'라 칭함)(340)는 7비트의 입력 신호 RR0=RR1를 입력하여 하기의 <수학식 2>에 적용함으로써 7비트의 Y0,Y1,...Y6신호를 출력한다.

<25>

$$
\begin{aligned}
y0 &= x1x3 \oplus x4 \oplus x0x1x4 \oplus x5 \oplus x2x5 \oplus x3x4x5 \oplus x6 \oplus x0x6 \oplus x1x6 \oplus x3x6 \oplus x2x4x6 \oplus x1x5x6 \\
&\quad \oplus x4x5x6 \\
y1 &= x0x1 \oplus x0x4 \oplus x2x4 \oplus x5 \oplus x1x2x5 \oplus x0x3x5 \oplus x6 \oplus x0x2x6 \oplus x3x6 \oplus x4x5x6 \oplus 1 \\
y2 &= x0 \oplus x0x3 \oplus x2x3 \oplus x1x2x4 \oplus x0x3x4 \oplus x1x5 \oplus x0x2x5 \oplus x0x6 \oplus x0x1x6 \oplus x2x6 \oplus x4x6 \oplus 1 \\
y3 &= x1 \oplus x0x1x2 \oplus x1x4 \oplus x3x4 \oplus x0x5 \oplus x0x1x5 \oplus x2x3x5 \oplus x1x4x5 \oplus x2x6 \oplus x1x3x6 \\
y4 &= x0x2 \oplus x3 \oplus x1x3 \oplus x1x4 \oplus x0x1x4 \oplus x2x3x4 \oplus x0x5 \oplus x1x3x5 \oplus x0x4x5 \oplus x1x6 \oplus x3x6 \\
&\quad \oplus x0x3x6 \oplus x5x6 \oplus 1 \\
y5 &= x2 \oplus x0x2 \oplus x0x3 \oplus x1x2x3 \oplus x0x2x4 \oplus x0x5 \oplus x2x5 \oplus x4x5 \oplus x1x6 \oplus x1x2x6 \oplus x0x3x6 \\
&\quad \oplus x3x4x6 \oplus x2x5x6 \oplus 1 \\
y6 &= x1x2 \oplus x0x1x3 \oplus x0x4 \oplus x1x5 \oplus x3x5 \oplus x6 \oplus x0x1x6 \oplus x2x3x6 \oplus x1x4x6 \oplus x0x5x6
\end{aligned}
$$

【수학식 2】

<26> 상기 TR1부(330)의 7비트 출력신호와 상기 S71연산기(340)의 7비트 출력신호는 서브 암호화키 KI1,1,1과 배타적 논리합 연산되어 7비트 RR2 신호로 출력된다.

<27> SBox92연산기(이하 'S92'라 칭함)(350)는 9비트의 입력 신호 RL2를 입력하여 상기의 <수학식 1>에 적용함으로써 9비트의 신호들 Y0,Y1,...Y8를 출력한다. ZE2부(360)는 7비트의 RR1신호를 입력받아 최상위비트(Most Significant Bit,이하 'MSB'라 칭함)에 2개의 ZERO(0)비트를 추가하여 9비트의 신호를 출력한다. 상기 S92연산기(350)의 9비트 출력신호와 상기 ZE2부(360)의 9비트 출력신호는 배타적 논리합 연산되어 9비트의 RL3신호로 출력한다. 상기 RL3신호는 TR2부(370)는 상기 9비트 RL3신호의 MSB 비트 중 ZERO(0)비트에 위치한 2개의 신호들을 제거하여 7비트의 신호를 출력한다. SBox72연산기(이하 'S72'라 칭함)(380)는 7비트의 입력 신호 RR2=RR3를 입력하여 상기의 <수학식 2>에 적용함으로써 7비트의 Y0,Y1,...Y6신호를 출력한다. 상기 TR2부(370)의 7비트 출력신호와 상기 S72연산기(380)의 7비트 출력신호는 배타적 논리합 연산되어 7비트 RR4신호로 출력된다.

<28> 따라서, 상기 FI1,1서브암호화부는 상기 9비트의 RL3=RL4 신호와 7비트의 RR4를 연산하여 16비트의 암호화된 신호(L4 ‖ R4)를 출력한다.

<29> 상기 전술한 바와 같이, 상기 S91연산기(310)와 S92연산기(350)는 상기 <수학식 1>에 의해 Y0, Y1,... Y8의 출력신호를 얻고자 논리곱 연산과 배타적 논리합 연산을 순차적으로 이룬다. 또한, 상기 S71연산기(340)와 상기 S72연산기(360)는 상기 <수학식 2>에 따라 Y0, Y1,... Y6의 출력 신호를 얻기 위해 논리곱 연산과 배타적 논리합 연산을 순차적으로 이루어진다. 즉, 입력된 신호는 상기 <수학식 1>에 따라 <수학식 2>에 의해 순차적으로 연산함에 따라 암호화에 따른 연산 처리 속도가 저하되는 문제점이 있었다.

또한, 상기 S91연산기(310), S92연산기(350), S71연산기(340), S72연산기(360)에서 논리 연산과정 중 발생하는 게이트 지연(gate delay)으로 인해 부정확한 암호화 신호(glitch)의 양이 점차적으로 증가하는 문제점이 있었다.

<30> 따라서, 본 발명은 KASUMI 알고리즘을 구현하는데 있어서, 정확한 신호의 연산과, 내부 연산처리 속도를 향상시키고, 지연되는 신호의 동기를 맞추기 위해 사용되는 부가적인 소자들을 절감시키는 KASUMI알고리즘을 새롭게 정의하고자 한다.

【발명이 이루고자 하는 기술적 과제】

<31> 따라서 상기한 바와 같이 동작되는 종래 기술의 문제점을 해결하기 위하여 창안된 본 발명의 목적은, 길이가 2n인 비트 열을 입력하여 길이가 2n인 암호화 비트 열로 출력하는 암호화 방법을 제공하는 것이다.

<32> 본 발명의 다른 목적은, 길이가 2n인 비트 열을 입력하여 길이가 2n인 암호화 비트 열로 출력하는 암호화 장치를 제공하는 것이다.

<33> 상기한 바와 같은 목적을 달성하기 위하여 창안된 본 발명의 실시예는, 길이가 2n인 제1입력 비트 열을 길이가 n인 제1 및 제2서브비트 열들로 분할하고, 길이 2n인 제2입력 비트 열을 길이 n인 제3 및 제4서브비트 열들로 분할하며, 상기 제1서브비트 열 내지 상기 제4서브비트 열 각각을 2단계의 암호화 절차에 의해 길이가 2n인 암호화 비트 열로 출력하는 암호화 방법에 있어서,

<34> 상기 제1서브비트 열과 상기 제2서브비트 열을 입력하여 상기 2단계의 암호화 절차 중 첫 번째 암호화를 위해 요구되는 소정 제1암호화 코드($KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, $KI_{1,2}$, $KI_{1,3}$)에 의해 암호화함으로써 길이가 n인 제1암호화 비트 열을 출력한 후 소정 지연 값을 가지고 제2암호화 비트 열을 출력하는 제1암호화 과정과,

<35> 상기 제1암호화 과정에 의해 출력되는 길이가 n인 제1암호화 비트 열을 상기 제3서브비트 열과 배타적 논리합 연산하여 제1연산 암호화 비트 열을 출력하고, 상기 제1암호화 과정에 의해 출력되는 길이가 n인 제2암호화 비트 열을 상기 제4서브비트 열과 배타적 논리합 연산하여 제2연산 암호화 비트 열을 출력하는 연산 과정과,

<36> 상기 제1연산 암호화 비트 열과 상기 소정 지연 값을 가지는 상기 제2연산 암호화 비트 열을 입력하여 상기 2단계의 암호화 절차 중 두 번째 암호화를 위해 요구되는 소정 제2암호화 코드($KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, $KI_{2,3}$)에 의해 암호화함으로써 동일 시점에서 각각의 길이가 n인 제3암호화 비트 열과 제4암호화 비트 열을 출력하는 제2암호화 과정을 포함함을 특징으로 한다.

<37> 상기한 바와 같은 목적을 달성하기 위하여 창안된 본 발명의 다른 실시예는, 길이가 2n인 제1입력 비트 열을 길이가 n인 제1 및 제2서브비트 열들로 분할하고, 길이 2n인 제2입력 비트 열을 길이 n인 제3 및 제4서브비트 열들로 분할하며, 상기 제1서브비트 열 내지 상기 제4서브비트 열 각각을 2단계의 암호화 절차에 의해 길이가 2n인 암호화 비트 열로 출력하는 암호화 장치에 있어서,

<38> 상기 제1서브비트 열과 상기 제2서브비트 열을 입력하여 상기 2단계의 암호화 절차 중 첫 번째 암호화를 위해 요구되는 소정 제1암호화 코드($KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, KI

$KI_{1,2}$, $KI_{1,3}$)에 의해 암호화함으로써 길이가 n인 제1암호화 비트 열을 출력한 후 소정 지연 값을 가지고 제2암호화 비트 열을 출력하는 제1암호화부와,

<39> 상기 제1암호화부에 의해 출력되는 길이가 n인 제1암호화 비트 열을 상기 제3서브비트 열과 배타적 논리합 연산하여 제1연산 암호화 비트 열을 출력하고, 상기 제1암호화 과정에 의해 출력되는 길이가 n인 제2암호화 비트 열을 상기 제4서브비트 열과 배타적 논리합 연산하여 제2연산 암호화 비트 열을 출력하는 연산부와,

<40> 상기 제1연산 암호화 비트 열과 상기 소정 지연 값을 가지는 상기 제2연산 암호화 비트 열을 입력하여 상기 2단계의 암호화 절차 중 두 번째 암호화를 위해 요구되는 소정 제2암호화 코드($KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, $KI_{2,3}$)에 의해 암호화함으로써 동일 시점에서 각각의 길이가 n인 제3암호화 비트 열과 제4암호화 비트 열을 출력하는 제2암호화부를 포함함을 특징으로 한다.

【발명의 구성 및 작용】

<41> 이하 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대한 동작 원리를 상세히 설명한다. 하기에서 본 발명을 설명함에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

<42> 본 발명에서 설명하고자 하는 KASUMI알고리즘은 암호화 및 무결성에서 사용되는 알고리즘으로, 암호화 알고리즘인 f8함수와 무결성 알고리즘인 f9함수에 사용되는 암호화 알고리즘이다. 상기 암호화 알고리즘 f8은 소정의 비트로 입력되는 평문신호를 암호화키와 배타적 논리합 연산하여 암호화하고, 상기 암호화된 암호문신호를 상기 암호화키와 배타적 논리합 연산하여 복호화하는 알고리즘이다. 또한, 무결성 알고리즘 f9는 수신된 신호로부터 메시지의 인증 코드를 유도하는 알고리즘이다. 상기 KASUMI알고리즘은 상기 설명한 바와 같이 암호화 및 무결성에 중요한 문제로 제기되고 있다.

<43> 도 4는 본 발명에서 제안하고 있는 KASUMI 알고리즘을 도시한 도면이다. 즉, 상기 도 4에서는, 64비트의 평문신호(plaintext)를 입력하고 제 1 암호화키, 제 2 암호화키, 제 3 암호화키에 의해 암호화하여 64비트의 암호화신호(ciphertext)를 출력한다. 즉, 상기 64비트의 평문신호(plaintext)는 복수 개의 FLi부($1 \leq i \leq 8$)들 (410,420,430,440,450,460,470,480)에서 해당하는 각각의 제 1 암호화 키 KLi($1 \leq i \leq 8$)와 복수 개의 SLIMFOi부($1 \leq I \leq 4$이고, 이하 '슬림FOi'라 칭함. 510,520,530,540)에서 해당하는 각각의 암호화 키 KOi($1 \leq i \leq 8$), KIi($1 \leq i \leq 8$)에 의해 암호화되어 64비트의 암호화 신호를 출력한다. 다음에서 상기 암호화 단계를 좀더 구체적으로 설명하고자 한다.

<44> 상기 64비트의 평문신호 중 분할된 32비트의 L0신호는 FL1부(410)에서 첫 번째 제 1암호화키 KL1에 의해 암호화되어 L1신호를 출력한다. 슬림FO1부(510)에서 상기 L1신호는 첫 번째 제 2 암호화키 KO1와 첫 번째 제 3 암호화 키 KI1에 의해 암호화된다. 상기 암호화된 L1신호는 32비트의 R0신호와 연산하여 SR1신호를 출력한다. 상기 SR1신호는 두 번째 제 2 암호화키 KO2과 두 번째 제 3 암호화 키 KI2에 의해 암호화되어 R1신호를 출력한다. 상기 R1신호는 FL2부(420)에서 두 번째 제 1암호화키 KL2에 의해 암호화되어 R2

신호를 출력한다. 상기 입력 신호 L0와 상기 R2신호는 배타적 논리합 연산되어 L2=SL1신호로 출력된다.

<45> 상기 L2=SL1신호는 FL3부(430)에서 세 번째 제 1암호화키 KL3에 의해 암호화되어 L3신호를 출력한다. 슬림F02부(520)에서 상기 L3신호는 세 번째 제 2 암호화키 K03과 세 번째 제 3 암호화 키 KI3에 의해 암호화된다. 상기 암호화된 L3신호는 상기 SR1신호와 연산하여 SR2신호를 출력한다. 상기 SR2신호는 네 번째 제 2 암호화키 K04와 네 번째 제 3 암호화 키 KI4에 의해 암호화되어 R3신호를 출력한다. 상기 R3신호는 FL4부(440)에서 네 번째 제 1암호화키 KL4에 의해 암호화되어 R4신호를 출력한다. 상기 R4신호는 상기 L2=SL1신호와 배타적 논리합 연산되어 L4=SL2신호로 출력된다.

<46> 상기 L4=SL2신호는 FL5부(450)에서 다섯 번째 제 1암호화키 KL5에 의해 암호화되어 L5 신호로 출력된다. 슬림F03부(530)에서 상기 L5신호는 다섯 번째 제 2 암호화키 K05와 다섯 번째 제 3 암호화 키 KI5에 의해 암호화된다. 암호화된 L5신호는 상기 SR2신호와 연산하여 SR3신호로 출력된다. 상기 SR3신호는 여섯 번째 제 2 암호화키 K06과 여섯 번째 제 3 암호화 키 KI6에 의해 암호화되어 R5신호를 출력한다. 상기 R5신호는 FL6부(460)에서 여섯 번째 제 1암호화키 KL6에 의해 암호화되어 R6 신호를 출력한다. 상기 R6 신호는 상기 L4=SL2신호와 배타적 논리합 연산하여 L6=SL3신호를 출력한다.

<47> 상기 L6=SL3신호는 FL7부(470)에서 일곱 번째 제 1암호화키 KL7에 의해 암호

화되어 L7신호로 출력된다. 슬림F04부(540)에서 상기 L7신호는 일곱 번째 제 2 암호화 키 KO7과 일곱 번째 제 3 암호화 키 KI7에 의해 암호화된다. 암호화된 L7신호는 상기 SR3신호와 연산하여 SR4신호로 출력된다. 상기 SR4신호는 여덟 번째 제 2 암호화키 KO8과 여덟 번째 제 3 암호화 키 KI8에 의해 암호화되어 R7신호를 출력한다. FL8부(480)에서 상기 R7신호는 여덟 번째 제 1암호화키 KL8에 의해 암호화되어 R8신호를 출력한다. 상기 R8 신호는 상기 L6=SL3와 배타적 논리합 연산하여 L8=SL4 신호를 출력한다. 따라서, 64비트의 평문신호는 8개의 FLi부($1 \leq i \leq 8$)들(410,420,430,440,450,460,470,480)와 4개의 슬림FOi부($1 \leq i \leq 4$)들(510,520,530,540)에 의해 암호화되어 64비트의 암호화신호(32비트 SL4신호 ‖ 32비트의 SR4신호)로 출력된다.

<48> 도 5는 본 발명에 따라 상기 도 4에서의 슬림FOi부 상세 구조를 도시한 도면이다.

<49> 상기 도 5를 참조하면, 상기 슬림FOi부는 i번째 슬림FO부를 말하는 것으로, 두 개의 FOi부를 병렬 연산하여 암호화하는 것이다. 여기서는 첫 번째 슬림FOi부를 예로 든다. 상기 슬림FO1부는 제 1암호화부인 FO1암호화부(501)와 제 2 암호화부 FO2암호화부(502)로 구성된다. 상기 제 1 암호화부 FO1암호화부(501)와 상기 제 2 암호화부 FO2암호화부(502)는 각각 FIi,j서브암호화부들($1 \leq i \leq 2$이고, $1 \leq j \leq 3$)에 의해 각각 3단계의 암호화 과정으로 이루어진다.

<50> 제 1 암호화부인 FO1암호화부(501)에서, 32비트의 입력신호는 상기 도 4에서 입력신호 64비트중 분할되어 입력된 32비트의 LO신호를 첫 번째 제 1암호화키 KL1에 의해 암호화된 신호이다. 상기 암호화된 신호는 16비트의 LO=L1신호와 16비트의 RO=R1신호로 분할되어 입력된다. 16비트의 LO=L1신호는 16 비트의 첫 번째 서브 암호화키 KO1,1에 의해 배타적 논리합 연산하여 L2신호를 출력한다. 상기 L2신호는 FI1,1서브암호화부(511)에

서 16비트의 첫 번째 서브 암호화키 KI1,1에 의해 암호화되어 지연된 L2D신호를 출력한다. 16비트의 RO=R1신호는 지연기D6(600)를 통과하면서 지연된 R1D신호를 출력한다. 상기 R1D신호는 상기 L1D신호는 배타적 논리합 연산하여 L3신호를 출력한다. 반면에 상기 RO=R1신호는 16 비트의 두 번째 서브 암호화키 KO1,2와 배타적 논리합 연산하여 R2신호를 출력한다. 상기 R2신호는 FI1,2서브암호화부(512)에서 16 비트의 두 번째 서브 암호화키 KI1,2에 의해 암호화되어 지연된 R2D 신호를 출력한다. 상기 R2D신호와 상기 L3신호는 배타적 논리합 연산하여 R3 신호를 출력한다. 또한, 상기 L3 신호는 16비트의 세 번째 서브 암호화키 KO1,3에 배타적 논리합 연산하여 L4신호를 출력한다. 상기 L4신호는 FI1,3서브암호화부(513)에서 16비트의 세 번째 서브 암호화키 KI1,3에 의해 암호화하여 지연된 L4D신호를 출력한다. 이때, 상기 R3신호는 지연기 D7(620)을 통과하면서 지연된 R3D신호를 출력한다. 상기 L4D신호는 상기 R3D신호와 배타적 논리합 연산을 하여 16비트의 L5신호를 출력한다.

<51> 제 2 암호화부인 FO2암호화부(502)에서, 입력신호는 상기 도 4에서 입력신호 64비트 중 분할되어 입력된 32비트의 RO신호이다. 상기 RO 신호는 16비트의 LO'신호와 16비트의 RO'신호로 분할되어 입력된다. 상기 LO'신호는 상기 16비트의 L5신호와 배타적 논리합 연산하여 L6신호를 출력한다. 반면에 상기 RO'신호는 상기 16비트의 R3신호와 배타적 논리합 연산하여 R4신호를 출력한다. 상기 R4신호는 16 비트의 첫 번째 서브 암호화키 KO2,1에 의해 배타적 논리합 연산하여 R5신호를 출력한다. 상기 R5신호는 FI2,1서브암호화부(514)에서 첫 번째 서브 암호화키 KI2,1에 의해 암호화되어 지연된 R5D신호를 출력한다. 상기 지연된 R5D신호는 상기 L6신호와 배타적 논리합 연산을 하여 신호 R6를 출력한다. 즉, 상기 FL1,3서브암호화부(513)와 상기 FL2,1서브암호화부(514)는 별도의

지연기를 사용하지 않고 상기 신호 L6와 R6신호의 동기를 일치시키는 역할을 한다. 상기 L6신호는 16 비트의 두 번째 서브 암호화키 KO2,2과 배타적 논리합 연산하여 L7신호를 출력한다. 상기 L7신호는 FI2,2서브암호화부(515)에서 16비트의 두 번째 서브 암호화키 KI2,2에 의해 암호화되어 지연된 L7D신호를 출력한다. 상기 R6신호는 지연기D8(640)을 통과하면서 지연된 R6D신호를 출력한다. 상기 L7D신호는 상기 R6D신호와 배타적 논리합 연산을 하여 L8신호를 출력한다. 상기 R6신호는 16비트의 세 번째 서브 암호화키 KO2,3과 배타적 논리합 연산하여 R7신호를 출력한다. 상기 R7신호는 FI2,3서브암호화부(516)에서 16 비트의 세 번째 서브 암호화키 KI2,3에 의해 지연된 R7D 신호를 출력한다. 상기 R7D신호는 상기 L8신호와 배타적 논리합 연산하여 R8신호를 출력한다. 따라서, 16비트 L8신호와 16비트 상기 R8신호 연산하여 32비트의 암호화 신호(L8‖R8)를 출력한다.

<52> 상기 전술한 바와 같이 상기 슬림FO1부는 상기 FO1암호화부(501)가 16비트의 LO신호와 16비트의 RO신호를 동시에 병렬 연산하고, 상기 FO2암호화부(502)가 16비트의 LO' 신호와 16비트의 RO'신호를 동시에 병렬 연산하여 암호화하는 것이다. 즉, 상기 복수개의 슬림FOi부는 64비트의 입력신호를 분할하고 32비트의 LO신호와 RO 신호를 동시에 병렬 연산하는 것으로, 암호화 처리 속도가 현저하게 증가하게 된다. 또한, 병렬 연산함으로써, 지연된 신호와 지연되지 않은 신호의 동기를 일치시키기 위하여 부가적으로 사용되는 지연기들을 감소시키는 이점이 있다.

<53> 도 6은 본 발명에 따라 상기 도 5에서의 FIi,j서브암호화부 상세 구조를 도시한 도면이다. 여기서는 상기 도 5에서 첫 번째 FI1,1서브암호화부(511)를 예를 들어 설명한다.

<54> 상기 도 6을 참조하면, 상기 첫 번째 FI1,1서브암호화부(511)는 제 1 암호화연산부와 제 2 암호화연산부로 구성된다. 제 1 암호화연산부에서, 16비트의 입력 신호는 9비트 RL0신호와 7비트 RR0신호로 분할된다. SBox91연산기(이하 'S91'라 칭함)(710)는 9비트의 입력 신호 RL0를 입력하여 하기의 <수학식 3>에 적용함으로써 9비트의 신호들 Y0,Y1,...Y8을 출력한다.

<55>

```
y0 = (x0x2)⊕x3⊕(x2x5)⊕(x5x6)⊕(x0x7)⊕(x1x7)⊕(x2x7)⊕(x4x8)⊕(x5x8)⊕(x7x8)⊕'1';
y1 = x1⊕(x0x1)⊕(x2x3)⊕(x0x4)⊕(x1x4)⊕(x0x5)⊕(x3x5)⊕x6⊕(x1x7)⊕(x2x7)⊕(x5x8)⊕'1';
y2 = x1⊕(x0x3)⊕(x3x4)⊕(x0x5)⊕(x2x6)⊕(x3x6)⊕(x5x6)⊕(x4x7)⊕(x5x7)⊕(x6x7)⊕x8⊕(x0x8)⊕'1';
y3 = x0⊕(x1x2)⊕(x0x3)⊕(x2x4)⊕x5⊕(x0x6)⊕(x1x6)⊕(x4x7)⊕(x0x8)⊕(x1x8)⊕(x7x8);
y4 = (x0x1)⊕(x1x3)⊕x4⊕(x0x5)⊕(x3x6)⊕(x0x7)⊕(x6x7)⊕(x1x8)⊕(x2x8)⊕(x3x8);
y5 = x2⊕(x1x4)⊕(x4x5)⊕(x0x6)⊕(x1x6)⊕(x3x7)⊕(x4x7)⊕(x6x7)⊕(x5x8)⊕(x6x8)⊕(x7x8)⊕'1';
y6 = x0⊕(x2x3)⊕(x1x5)⊕(x2x5)⊕(x4x5)⊕(x3x6)⊕(x4x6)⊕(x5x6)⊕x7⊕(x1x8)⊕(x3x8)⊕(x5x8)⊕(x7x8);
y7 = (x0x1)⊕(x0x2)⊕(x1x2)⊕x3⊕(x0x3)⊕(x2x3)⊕(x4x5)⊕(x2x6)⊕(x3x6)⊕(x2x7)⊕(x5x7)⊕x8⊕'1';
y8 = (x0x1)⊕x2⊕(x1x2)⊕(x3x4)⊕(x1x5)⊕(x2x5)⊕(x1x6)⊕(x4x6)⊕x7⊕(x2x8)⊕(x3x8);
```

【수학식 3】

<56> 즉, 상기 S91연산기(710)는 9비트 신호 X0,X1,....X8을 입력하여 논리곱(and 게이트 연산)에 괄호를 적용하여 병렬 연산하고, 배타적 논리합 연산도 병렬 연산하여 9비트 신호 Y0,Y1,...Y8를 출력한다. ZE1부(720)는 7비트의 RR0신호를 입력받아 최상위비트(Most Significant Bit,이하 'MSB'라 칭함)에 2개의 ZERO(0)비트를 추가하여 9비트의 신호를 출력한다. 상기 S91연산기수(710)의 9비트 출력신호는 상기 ZE1부(720)의 9비트 출력신호와 배타적 논리합 연산되어 9비트의 RL1신호로 출력한다. 상기 RL1신호는 9비트의 서브 암호화키 KI1,1,2와 배타적 논리합 연산하여 9비트의 RL2신호를 출력한다. 상기 9비트의 RL2신호는 레지스터 1(800)에 임시 저장된다.

<57> 동시에, SBox71연산기(이하 'S71'라 칭함)(740)는 7비트의 입력 신호 RR0=RR1를 입력하여 하기의 <수학식 4>에 적용함으로써 7비트의 Y0,Y1,...Y6신호를 출력한다.

<58>

$$y0=(x1x3)\oplus x4\oplus(x0x1x4)\oplus x5\oplus(x2x5)\oplus(x3x4x5)\oplus x6\oplus(x0x6)\oplus(x1x6)\oplus(x3x6)\oplus(x2x4x6)\oplus(x1x5x6)\oplus(x4x5x6);$$

$$y1=(x0x1)\oplus(x0x4)\oplus(x2x4)\oplus x5\oplus(x1x2x5)\oplus(x0x3x5)\oplus x6\oplus(x0x2x6)\oplus(x3x6)\oplus(x4x5x6)\oplus 1';$$

$$y2=x0\oplus(x0x3)\oplus(x2x3)\oplus(x1x2x4)\oplus(x0x3x4)\oplus(x1x5)\oplus(x0x2x5)\oplus(x1x6)\oplus(x0x4x6)\oplus(x2x6)\oplus(x4x6)\oplus 1';$$

$$y3=x1\oplus(x0x1x2)\oplus(x1x4)\oplus(x3x4)\oplus(x0x5)\oplus(x0x1x5)\oplus(x2x3x5)\oplus(x1x4x5)\oplus(x2x6)\oplus(x1x3x6);$$

$$y4=(x0x2)\oplus x3\oplus(x1x3)\oplus(x1x4)\oplus(x0x1x4)\oplus(x2x3x4)\oplus(x0x5)\oplus(x1x3x5)\oplus(x0x4x5)\oplus(x1x6)\oplus(x3x6)\oplus(x0x3x6)\oplus(x5x6)\oplus 1';$$

$$y5=x2\oplus(x0x2)\oplus(x0x3)\oplus(x1x2x3)\oplus(x0x2x4)\oplus(x0x5)\oplus(x2x5)\oplus(x4x5)\oplus(x1x6)\oplus(x1x2x6)\oplus(x0x3x6)\oplus(x3x4x6)\oplus(x2x5x6)\oplus 1';$$

$$y6=(x1x2)\oplus(x0x1x3)\oplus(x0x4)\oplus(x1x5)\oplus(x3x5)\oplus x6\oplus(x0x1x6)\oplus(x2x3x6)\oplus(x1x4x6)\oplus(x0x5x6);$$

【수학식 4】

<59> 즉, 상기 S71연산기(740)는 7비트 신호 X0,X1,....X8을 입력하여 논리곱(and 게이트 연산)에 괄호를 적용하여 병렬 연산하고, 배타적 논리합 연산도 병렬 연산하여 7비트 신호 Y0,Y1,...Y6를 출력한다. TR1부(730)는 상기 9비트 RL1신호의 MSB 비트 중 ZERO(0)비트에 위치한 2개의 신호들을 제거하여 7비트의 신호를 출력한다. 상기 TR1부(730)의 7비트 출력 신호와 상기 S71연산기(740)의 7비트 출력신호는 서브 암호화키 KI1,1,1과 배타적 논리합 연산하여 7비트 RR2 신호를 출력한다. 상기 7비트의 RR2 신호는 레지스터 1(800)에 임시 저장된다. 상기 9비트의 RL2신호와 상기 7비트의 RR2신호를 저장한 상기 레지스터1(800)는 제어부(본 명세서에서는 도시하지 않음)로부터 클럭 1(CLK1)신호가 인가되면, 동시에 각각 9비트 RL2신호와 7비트 RR2신호를 출력된다. 따라서, 상기 레지스터1(800)은 상기 S91연산기(710), ZE1부(720), TR1부(730), S71연산기(740)의 암호화과정에서 발생하는 내부지연에 따른 출력 동기를 맞추는 역할을 한다.

<60> 제 2 암호화연산부에서, SBox92연산기(이하 'S92'라 칭함)(750)은 상기 레지스터 1(800)로부터 출력된 9비트의 RL2를 입력하여 상기의 <수학식 3>에 적용함으로써 9비트의 신호들 Y0,Y1,...Y8를 출력한다. ZE2부(760)는 상기 레지스터1(800)로부터 출력된 7비트의 RR2신호를 입력받아 최상위비트(Most Significant Bit,이하 'MSB'라 칭함)에 2개의 ZERO(0)비트를 추가하여 9비트의 신호를 출력한다. 상기 S92연산기(750)의 9비트 출

력신호와 상기 ZE2부(760)의 9비트 출력신호는 배타적 논리합 연산되어 9비트의 RL3신호로 출력된다. 상기 RL3=RR4신호는 레지스터 2(820)에 임시 저장된다.

<61> 동시에, SBox72연산기(이하 'S72'라 칭함)(780)는 7비트의 입력 신호 RR2=RR3를 입력하여 상기의 <수학식 4>에 적용함으로써 7비트의 Y0,Y1,...Y6신호를 출력한다. 상기 RL3신호는 TR2부(770)에서 상기 9비트 RL3신호의 MSB 비트 중 ZERO(0)비트에 위치한 2개의 신호들을 제거하여 7비트의 신호를 출력한다. 상기 TR2부(770)의 7비트 출력 신호와 상기 S72연산기(780)의 7비트 출력신호는 배타적 논리합 연산하여 7비트 RR4 신호를 출력한다. 상기 7비트의 RR4 신호는 레지스터 2(820)에 임시 저장된다.

<62> 상기 9비트의 RL4신호와 상기 7비트의 RR4신호를 저장한 상기 레지스터2(820)은 제어부(본 명세서에서는 도시하지 않음)로부터 클럭2(CLK2)신호가 인가되면, 동시에 각각 9비트 RL4신호와 7비트 RR4신호를 출력된다. 따라서, 상기 레지스터2(820)은 상기 S92연산기(750), ZE2부(760), TR2부(770), S72연산기(780)의 암호화과정에서 발생하는 내부지연에 따른 출력 동기를 맞추는 역할을 한다.

<63> 상기 전술한 바와 같이, 상기 S91연산기(710)와 S92연산기(750)는 새롭게 정의된 상기의 <수학식 3>에 따라 논리곱을 병렬 연산하고, 배타적 논리합 연산하여 9비트 신호 Y0,Y1,...Y8를 출력한다. 또한, 상기 S71연산기(740)와 상기 S72연산기(780)는 상기 <수학식 4>에 의해 논리곱을 병렬 연산하고, 배타적 논리합 연산하여 7비트의 Y0, Y1,... Y6의 출력한다. 즉, 입력신호들을 상기 <수학식 3>과 <수학식 4>에 적용함으로써, 논리곱 연산이 병렬 연산되어 암호화 연산 처리 속도가 현저하게 증가하게 된다. 또한, 상기 레지스터1(800)와 상기 레지스터2(820)를 사용하여 출력 동기를 맞추므로 정확한 암호화 신호를 출력하게 된다.

<64> 한편 본 발명의 상세한 설명에서는 구체적인 실시예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 그러므로 본 발명의 범위는 설명된 실시예에 국한되지 않으며, 후술되는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

【발명의 효과】

<65> 이상에서 상세히 설명한 바와 같이 동작하는 본 발명에 있어서, 개시되는 발명중 대표적인 것에 의하여 얻어지는 효과를 간단히 설명하면 다음과 같다.

<66> 본 발명은, 입력되는 신호들을 병렬 연산하여 내부 신호 처리 속도에 효율성을 가지는 효과가 있다. 또한, 내부 지연에 따른 지연된 신호와 지연이 발생하지 않은 신호의 출력 동기를 맞추어 최종적으로 정확한 신호를 출력하여 암호화 시스템의 안정성을 증가시키는 효과가 있다. 또한, 내부 지연에 따른 지연 동기를 맞추기 위해 사용되는 소자의 현저하게 감소하여 하드웨어 구성에 따른 칩용량이 축소되는 효과가 있다. 따라서, 적은 소자 사용으로 생산비용을 절감하는 효과가 있다.

【특허청구범위】

【청구항 1】

길이가 2n인 제1입력 비트 열을 길이가 n인 제1 및 제2서브비트 열들로 분할하고, 길이 2n인 제2입력 비트 열을 길이 n인 제3 및 제4서브비트 열들로 분할하며, 상기 제1서브비트 열 내지 상기 제4서브비트 열 각각을 2단계의 암호화 절차에 의해 길이가 2n인 암호화 비트 열로 출력하는 암호화 방법에 있어서,

상기 제1서브비트 열과 상기 제2서브비트 열을 입력하여 상기 2단계의 암호화 절차 중 첫 번째 암호화를 위해 요구되는 소정 제1암호화 코드($KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, $KI_{1,2}$, $KI_{1,3}$)에 의해 암호화함으로써 길이가 n인 제1암호화 비트 열을 출력한 후 소정 지연 값을 가지고 제2암호화 비트 열을 출력하는 제1암호화 과정과,

상기 제1암호화 과정에 의해 출력되는 길이가 n인 제1암호화 비트 열을 상기 제3서브비트 열과 배타적 논리합 연산하여 제1연산 암호화 비트 열을 출력하고, 상기 제1암호화 과정에 의해 출력되는 길이가 n인 제2암호화 비트 열을 상기 제4서브비트 열과 배타적 논리합 연산하여 제2연산 암호화 비트 열을 출력하는 연산 과정과,

상기 제1연산 암호화 비트 열과 상기 소정 지연 값을 가지는 상기 제2연산 암호화 비트 열을 입력하여 상기 2단계의 암호화 절차 중 두 번째 암호화를 위해 요구되는 소정 제2암호화 코드($KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, $KI_{2,3}$)에 의해 암호화함으로써 동일 시점에서 각각의 길이가 n인 제3암호화 비트 열과 제4암호화 비트 열을 출력하는 제2암호화 과정을 포함함을 특징으로 하는 상기 방법.

【청구항 2】

제1항에 있어서, 상기 제1암호화 과정은,

상기 제1서브비트 열을 상기 제1암호화 코드 중 $KO_{1,1}$과 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1암호화 코드 중 $KI_{1,1}$에 의해 제1암호화한 후 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제2서브 비트 열을 배타적 논리 합 연산하여 제1신호를 출력하는 단계와,

상기 제2서브비트 열을 상기 제1암호화 코드 중 $KO_{1,2}$와 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1암호화 코드 중 $KI_{1,2}$에 의해 제2암호화한 후 상기 제1신호와 배타적 논리 합 연산하여 상기 제1연산 암호화 비트 열을 출력하는 단계와,

상기 제1신호를 상기 제1암호화 코드 중 $KO_{1,3}$과 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1암호화 코드 중 $KI_{1,3}$에 의해 제3 암호화한 후 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제1연산 암호화 비트 열을 배타적 논리 합 연산하여 상기 제2연산 암호화 비트 열을 출력하는 단계를 포함함을 특징으로 하는 상기 방법.

【청구항 3】

제1항에 있어서, 상기 제2암호화 과정은,

상기 제1연산 암호화 비트 열을 상기 제2암호화 코드 중 $KO_{2,1}$과 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중

$KI_{2,1}$에 의해 제4암호화한 후 상기 제2연산 암호화 비트 열과 배타적 논리 합 연산하여 제2신호를 출력하는 단계와,

상기 제2연산 암호화 비트 열을 상기 제2암호화 코드 중 $KO_{2,2}$와 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중 $KI_{2,2}$에 의해 제5암호화한 후 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제2신호와 배타적 논리 합 연산하여 제3암호화 비트 열을 출력하는 단계와,

상기 제2신호를 상기 제2암호화 코드 중 $KO_{2,3}$과 배타적 논리 합 연산하고, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중 $KI_{2,3}$에 의해 제6암호화한 후 상기 제3암호화 비트 열과 배타적 논리 합 연산하여 상기 제4암호화 비트 열을 출력하는 단계를 포함함을 특징으로 하는 상기 방법.

【청구항 4】

제3항에 있어서, 상기 제1 내지 제6암호화는, 첫 번째 서브 암호화 단계와 두 번째 서브 암호화 단계로 이루어지며, 상기 첫 번째 서브 암호화 단계로부터의 출력들과 상기 두 번째 서브 암호화 단계의 출력들을 저장한 후 외부로부터의 클럭 신호에 의해 동시에 출력함을 특징으로 하는 상기 방법.

【청구항 5】

제4항에 있어서, 상기 첫 번째 서브 암호화 단계와 상기 두 번째 서브 암호화 단계는 길이가 16인 비트 열을 입력으로 하여 길이가 9인 비트 열과 길이가 7비트인 비트 열

로 분할하고, 상기 길이가 9비트인 비트 열을 하기 <수학식 5>에 적용함으로서 길이가 9비트인 암호화 비트 열을 출력하며, 상기 길이가 7비트인 비트 열을 하기 <수학식 6>에 적용함으로서 길이가 7비트인 암호화 비트 열을 출력하는 것을 적어도 포함함을 특징으로 하는 상기 방법.

$$
\begin{aligned}
y0 &= (x0x2)\oplus x3\oplus(x2x5)\oplus(x5x6)\oplus(x0x7)\oplus(x1x7)\oplus(x2x7)\oplus(x4x8)\oplus(x5x8)\oplus(x7x8)\oplus'1';\\
y1 &= x1\oplus(x0x1)\oplus(x2x3)\oplus(x0x4)\oplus(x1x4)\oplus(x0x5)\oplus(x3x5)\oplus x6\oplus(x1x7)\oplus(x2x7)\oplus(x5x8)\oplus'1';\\
y2 &= x1\oplus(x0x3)\oplus(x3x4)\oplus(x0x5)\oplus(x2x6)\oplus(x3x6)\oplus(x5x6)\oplus(x4x7)\oplus(x5x7)\oplus(x6x7)\oplus x8\oplus(x0x8)\oplus'1';\\
y3 &= x0\oplus(x1x2)\oplus(x0x3)\oplus(x2x4)\oplus x5\oplus(x0x6)\oplus(x1x6)\oplus(x4x7)\oplus(x0x8)\oplus(x1x8)\oplus(x7x8);\\
y4 &= (x0x1)\oplus(x1x3)\oplus x4\oplus(x0x5)\oplus(x3x6)\oplus(x0x7)\oplus(x6x7)\oplus(x1x8)\oplus(x2x8)\oplus(x3x8);\\
y5 &= x2\oplus(x1x4)\oplus(x4x5)\oplus(x0x6)\oplus(x1x6)\oplus(x3x7)\oplus(x4x7)\oplus(x6x7)\oplus(x5x8)\oplus(x6x8)\oplus(x7x8)\oplus'1';\\
y6 &= x0\oplus(x2x3)\oplus(x1x5)\oplus(x2x5)\oplus(x4x5)\oplus(x3x6)\oplus(x4x6)\oplus(x5x6)\oplus x7\oplus(x1x8)\oplus(x3x8)\oplus(x5x8)\oplus(x7x8);\\
y7 &= (x0x1)\oplus(x0x2)\oplus(x1x2)\oplus x3\oplus(x0x3)\oplus(x2x3)\oplus(x4x5)\oplus(x2x6)\oplus(x3x6)\oplus(x2x7)\oplus(x5x7)\oplus x8\oplus'1';\\
y8 &= (x0x1)\oplus x2\oplus(x1x2)\oplus(x3x4)\oplus(x1x5)\oplus(x2x5)\oplus(x1x6)\oplus(x4x6)\oplus x7\oplus(x2x8)\oplus(x3x8);
\end{aligned}
$$

【수학식 5】

$$
\begin{aligned}
y0&=(x1x3)\oplus x4\oplus(x0x1x4)\oplus x5\oplus(x2x5)\oplus(x3x4x5)\oplus x6\oplus(x0x6)\oplus(x1x6)\oplus(x3x6)\oplus(x2x5x6)\oplus(x1x5x6)\oplus(x4x5x6);\\
y1&=(x0x1)\oplus(x0x4)\oplus(x2x4)\oplus x5\oplus(x1x2x5)\oplus(x0x3x5)\oplus x6\oplus(x0x2x6)\oplus(x3x6)\oplus(x4x5x6)\oplus'1';\\
y2&=x0\oplus(x0x3)\oplus(x2x3)\oplus(x1x2x4)\oplus(x0x3x4)\oplus(x1x5)\oplus(x0x2x5)\oplus(x0x6)\oplus(x0x4x6)\oplus(x2x6)\oplus(x4x5)\oplus'1';\\
y3&=x1\oplus(x0x1x2)\oplus(x1x4)\oplus(x3x4)\oplus(x0x5)\oplus(x0x1x5)\oplus(x2x3x5)\oplus(x1x4x5)\oplus(x2x6)\oplus(x1x3x6);\\
y4&=(x0x2)\oplus x3\oplus(x1x3)\oplus(x1x4)\oplus(x0x1x4)\oplus(x2x3x4)\oplus(x0x5)\oplus(x1x3x5)\oplus(x0x4x5)\oplus(x1x6)\oplus(x3x5)\oplus(x0x3x6)\oplus(x5x6)\oplus'1';\\
y5&=x2\oplus(x0x2)\oplus(x0x3)\oplus(x1x2x3)\oplus(x0x2x4)\oplus(x0x5)\oplus(x2x5)\oplus(x4x5)\oplus(x1x6)\oplus(x1x2x6)\oplus(x0x3x6)\oplus(x3x4x6)\oplus(x2x5x6)\oplus'1';\\
y6&=(x1x2)\oplus(x0x1x3)\oplus(x0x4)\oplus(x1x5)\oplus(x3x5)\oplus x6\oplus(x0x1x6)\oplus(x2x3x6)\oplus(x1x4x6)\oplus(x0x5x6);
\end{aligned}
$$

【수학식 6】

【청구항 6】

길이가 2n인 제1입력 비트 열을 길이가 n인 제1 및 제2서브비트 열들로 분할하고, 길이 2n인 제2입력 비트 열을 길이 n인 제3 및 제4서브비트 열들로 분할하며, 상기 제1

서브비트 열 내지 상기 제4서브비트 열 각각을 2단계의 암호화 절차에 의해 길이가 2n인 암호화 비트 열로 출력하는 암호화 장치에 있어서,

상기 제1서브비트 열과 상기 제2서브비트 열을 입력하여 상기 2단계의 암호화 절차 중 첫 번째 암호화를 위해 요구되는 소정 제1암호화 코드($KO_{1,1}$, $KO_{1,2}$, $KO_{1,3}$, $KI_{1,1}$, $KI_{1,2}$, $KI_{1,3}$)에 의해 암호화함으로써 길이가 n인 제1암호화 비트 열을 출력한 후 소정 지연 값을 가지고 제2암호화 비트 열을 출력하는 제1암호화부와,

상기 제1암호화부에 의해 출력되는 길이가 n인 제1암호화 비트 열을 상기 제3서브비트 열과 배타적 논리합 연산하여 제1연산 암호화 비트 열을 출력하고, 상기 제1암호화 과정에 의해 출력되는 길이가 n인 제2암호화 비트 열을 상기 제4서브비트 열과 배타적 논리합 연산하여 제2연산 암호화 비트 열을 출력하는 연산부와,

상기 제1연산 암호화 비트 열과 상기 소정 지연 값을 가지는 상기 제2연산 암호화 비트 열을 입력하여 상기 2단계의 암호화 절차 중 두 번째 암호화를 위해 요구되는 소정 제2암호화 코드($KO_{2,1}$, $KO_{2,2}$, $KO_{2,3}$, $KI_{2,1}$, $KI_{2,2}$, $KI_{2,3}$)에 의해 암호화함으로써 동일 시점에서 각각의 길이가 n인 제3암호화 비트 열과 제4암호화 비트 열을 출력하는 제2암호화부를 포함함을 특징으로 하는 상기 장치.

【청구항 7】

제6항에 있어서, 상기 제1암호화부는,

상기 제1서브비트 열을 상기 제1암호화 코드 중 $KO_{1,1}$과 배타적 논리 합 연산하는 배타적 논리 합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1

암호화 코드 중 $KI_{1,1}$에 의해 암호화하는 제1서브 암호화부와, 상기 제1서브 암호화부로부터의 출력을 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제2서브비트 열과 배타적 논리 합 연산하여 제1신호를 출력하는 배타적 논리 합 연산자를 가지는 제1블록과,

상기 제2서브비트 열을 상기 제1암호화 코드 중 $KO_{1,2}$와 배타적 논리 합 연산하는 배타적 논리 합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1암호화 코드 중 $KI_{1,2}$에 의해 암호화하는 제2서브 암호화부와, 상기 제2서브 암호화부로부터의 출력을 상기 제1신호와 배타적 논리 합 연산하여 상기 제1연산 암호화 비트 열을 출력하는 배타적 논리합 연산자를 가지는 제2블록과,

상기 제1신호를 상기 제1암호화 코드 중 $KO_{1,3}$과 배타적 논리 합 연산하는 배타적 논리합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제1암호화 코드 중 $KI_{1,3}$에 의해 암호화하는 제3서브 암호화부와, 상기 제3서브 암호화부로부터의 출력을 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제1연산 암호화 비트 열과 배타적 논리 합 연산하여 상기 제2연산 암호화 비트 열을 출력하는 배타적 논리 합 연산자를 가지는 제3블록을 포함함을 특징으로 하는 상기 장치.

【청구항 8】

제6항에 있어서, 상기 제2암호화부는,

상기 제1연산 암호화 비트 열을 상기 제2암호화 코드 중 $KO_{2,1}$과 배타적 논리 합 연산하는 배타적 논리 합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중 $KI_{2,1}$에 의해 암호화하는 제4서브 암호화부와, 상기 제4암호화

부로부터의 출력을 상기 제2연산 암호화 비트 열과 배타적 논리 합 연산하여 제2신호를 출력하는 배타적 논리 합 연산자를 가지는 제4블록과,

상기 제2연산 암호화 비트 열을 상기 제2암호화 코드 중 $KO_{2,2}$와 배타적 논리 합 연산하는 배타적 논리 합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중 $KI_{2,2}$에 의해 암호화하는 제5서브 암호화부와, 상기 제5서브 암호화부로부터의 출력을 상기 암호화에 따른 처리 시간만큼이 지연된 상기 제2신호와 배타적 논리 합 연산하여 제3암호화 비트 열을 출력하는 배타적 논리 합 연산자를 가지는 제5블록과,

상기 제2신호를 상기 제2암호화 코드 중 $KO_{2,3}$과 배타적 논리 합 연산하는 배타적 논리 합 연산자와, 상기 배타적 논리 합 연산에 따른 출력 비트 열을 상기 제2암호화 코드 중 $KI_{2,3}$에 의해 암호화하는 제6서브 암호화부와, 상기 제6서브 암호화부로부터의 출력을 상기 제3암호화 비트 열과 배타적 논리 합 연산하여 상기 제4암호화 비트 열을 출력하는 배타적 논리 합 연산자를 가지는 제6블록을 포함함을 특징으로 하는 상기 장치.

【청구항 9】

제8항에 있어서, 제1 내지 제6서브 암호화부들 각각은 제 1 암호화연산부와 제 2 암호화연산부로 이루어지며, 상기 제 1 암호화연산부의 출력들과 상기 제 2 암호화연산부의 출력들을 저장한 후 외부로부터의 클럭 신호에 의해 동시에 출력하는 레지스터를 포함함을 특징으로 하는 상기 장치.

【청구항 10】

제9항에 있어서, 상기 제 1 암호화연산부와 상기 제 2 암호화연산부는 길이가 16인 비트 열을 입력으로 하여 길이가 9인 비트 열과 길이가 7비트인 비트 열로 분할하고, 상기 길이가 9비트인 비트 열을 하기 <수학식 7>에 적용함으로서 길이가 9비트인 암호화 비트 열을 출력하며, 상기 길이가 7비트인 비트 열을 하기 <수학식 8>에 적용함으로서 길이가 7비트인 암호화 비트 열을 출력하는 것을 적어도 포함함을 특징으로 하는 상기 장치.

```
y0 = (x0x2)⊕x3⊕(x2x5)⊕(x5x6)⊕(x0x7)⊕(x1x7)⊕(x2x7)⊕(x4x8)⊕(x5x8)⊕(x7x8)⊕'1';
y1 = x1⊕(x0x1)⊕(x2x3)⊕(x0x4)⊕(x1x4)⊕(x0x5)⊕(x3x5)⊕x6⊕(x1x7)⊕(x2x7)⊕(x5x8)⊕'1';
y2 = x1⊕(x0x3)⊕(x3x4)⊕(x0x5)⊕(x2x6)⊕(x3x6)⊕(x5x6)⊕(x4x7)⊕(x5x7)⊕(x6x7)⊕x8⊕(x0x8)⊕'1';
y3 = x0⊕(x1x2)⊕(x0x3)⊕(x2x4)⊕x5⊕(x0x6)⊕(x1x6)⊕(x4x7)⊕(x0x8)⊕(x1x8)⊕(x7x8);
y4 = (x0x1)⊕(x1x3)⊕x4⊕(x0x5)⊕(x3x6)⊕(x0x7)⊕(x6x7)⊕(x1x8)⊕(x2x8)⊕(x3x8);
y5 = x2⊕(x1x4)⊕(x4x5)⊕(x0x6)⊕(x1x6)⊕(x3x7)⊕(x4x7)⊕(x6x7)⊕(x5x8)⊕(x6x8)⊕(x7x8)⊕'1';
y6 = x0⊕(x2x3)⊕(x1x5)⊕(x2x5)⊕(x4x5)⊕(x3x6)⊕(x4x6)⊕(x5x6)⊕x7⊕(x1x8)⊕(x3x8)⊕(x5x8)⊕(x7x8);
y7 = (x0x1)⊕(x0x2)⊕(x1x2)⊕x3⊕(x0x3)⊕(x2x3)⊕(x4x5)⊕(x2x6)⊕(x3x6)⊕(x2x7)⊕(x5x7)⊕x8⊕'1';
y8 = (x0x1)⊕x2⊕(x1x2)⊕(x3x4)⊕(x1x5)⊕(x2x5)⊕(x1x6)⊕(x4x6)⊕x7⊕(x2x8)⊕(x3x8);
```

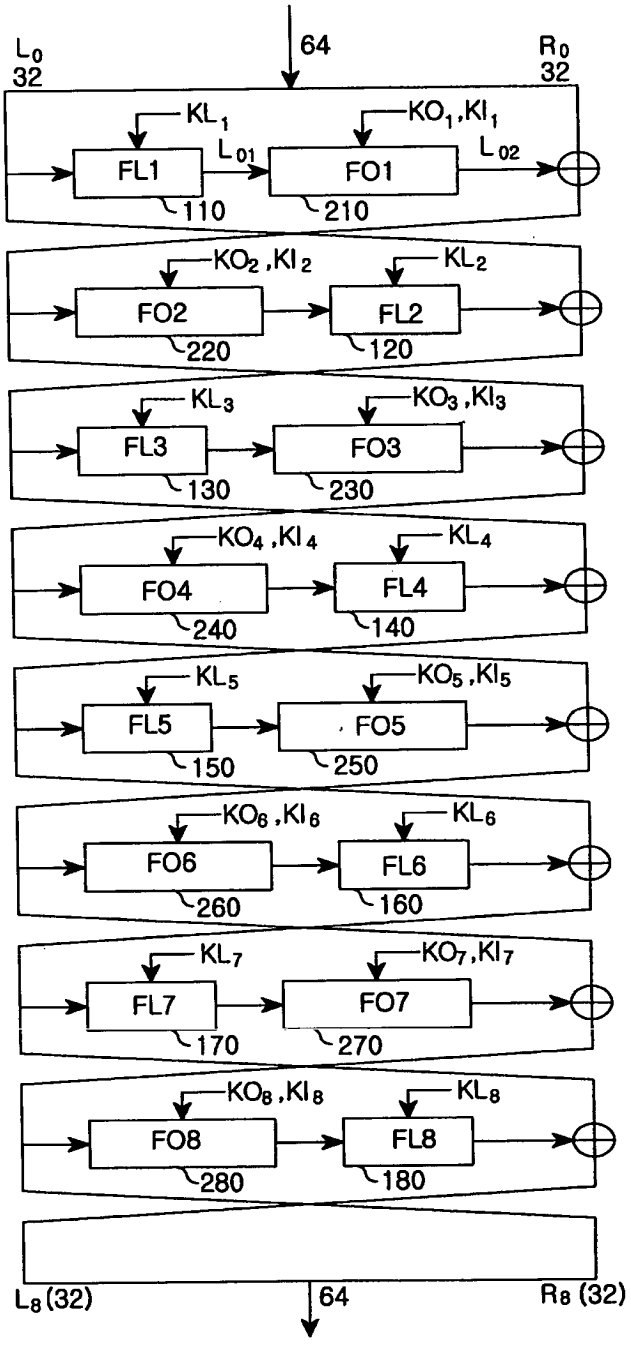
【수학식 7】

```
y0=(x1x2)⊕x4⊕(x0x1x4)⊕x5⊕(x2x5)⊕(x3x4x5)⊕x6⊕(x0x6)⊕(x1x6)⊕(x3x6)⊕(x2x4x6)⊕(x1x5x6)⊕(x4x5x6);
y1=(x0x1)⊕(x0x4)⊕(x2x4)⊕x5⊕(x1x2x5)⊕(x0x3x5)⊕x6⊕(x0x2x6)⊕(x3x6)⊕(x4x5x6)⊕'1';
y2=x0⊕(x0x3)⊕(x2x3)⊕(x1x2x4)⊕(x0x3x4)⊕(x1x5)⊕(x0x2x5)⊕(x0x6)⊕(x0x4x6)⊕(x2x6)⊕(x4x6)⊕'1';
y3=x1⊕(x0x1x2)⊕(x1x4)⊕(x3x4)⊕(x0x5)⊕(x0x1x5)⊕(x2x3x5)⊕(x1x4x5)⊕(x2x6)⊕(x1x3x6);
y4=(x0x2)⊕x3⊕(x1x3)⊕(x1x4)⊕(x0x1x4)⊕(x2x3x4)⊕(x0x5)⊕(x1x3x5)⊕(x0x4x5)⊕(x1x6)⊕(x3x6)⊕(x0x3x6)⊕(x5x6)⊕'1';
y5=x2⊕(x0x2)⊕(x0x3)⊕(x1x2x3)⊕(x0x2x4)⊕(x0x5)⊕(x2x5)⊕(x4x5)⊕(x1x6)⊕(x1x2x6)⊕(x0x3x6)⊕(x3x4x6)⊕(x2x5x6)⊕'1';
y6=(x1x2)⊕(x0x1x3)⊕(x1x4)⊕(x1x5)⊕(x3x5)⊕x6⊕(x0x1x6)⊕(x2x3x6)⊕(x1x4x6)⊕(x1x5x6);
```

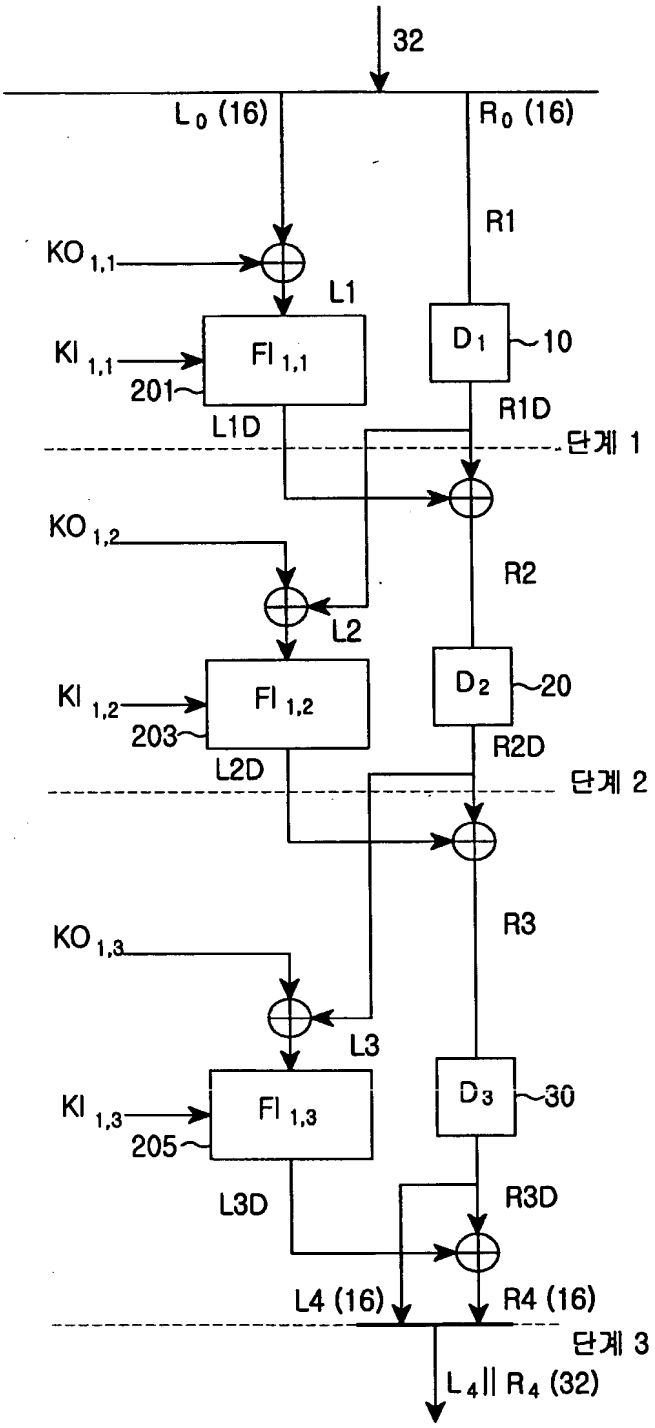
【수학식 8】

【도면】

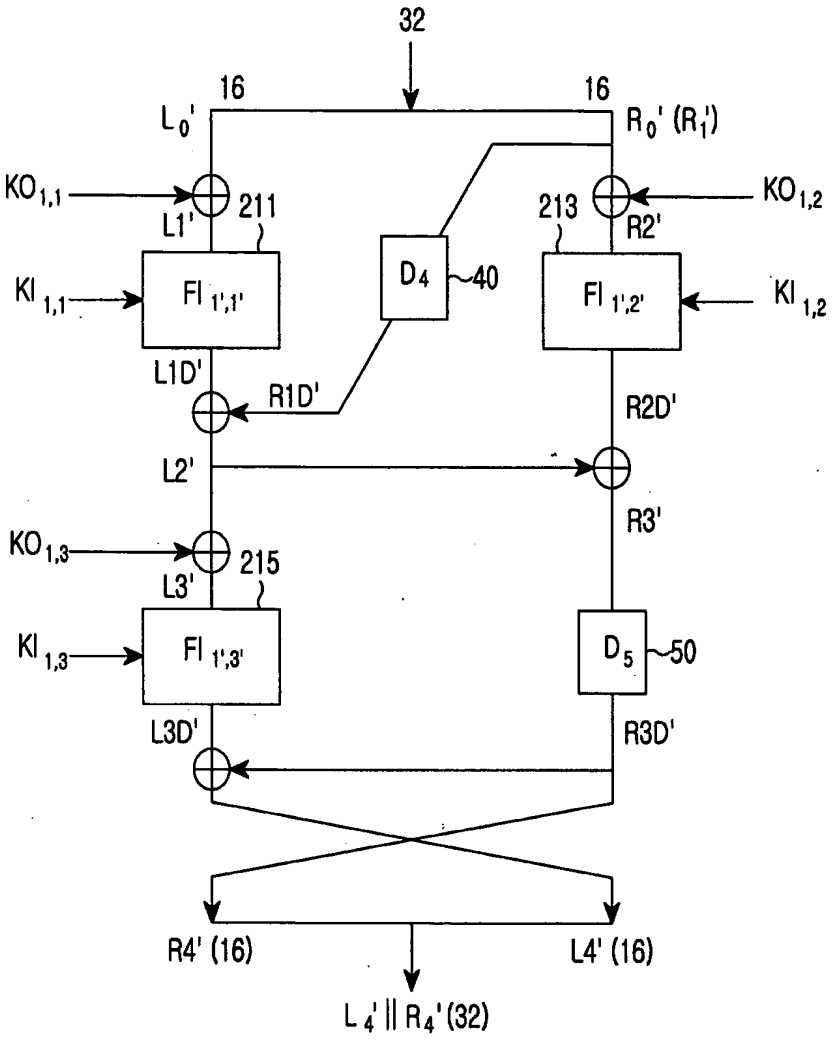
【도 1】


L0
32
64
R0
32
KL1
KO1,KI1
L01
L02
FL1
FO1
110
210
KO2,KI2
KL2
FO2
FL2
220
120
KL3
KO3,KI3
FL3
FO3
130
230
KO4,KI4
KL4
FO4
FL4
240
140
KL5
KO5,KI5
FL5
FO5
150
250
KO6,KI6
KL6
FO6
FL6
260
160
KL7
KO7,KI7
FL7
FO7
170
270
KO8,KI8
KL8
FO8
FL8
280
180
L8(32)
64
R8 (32)

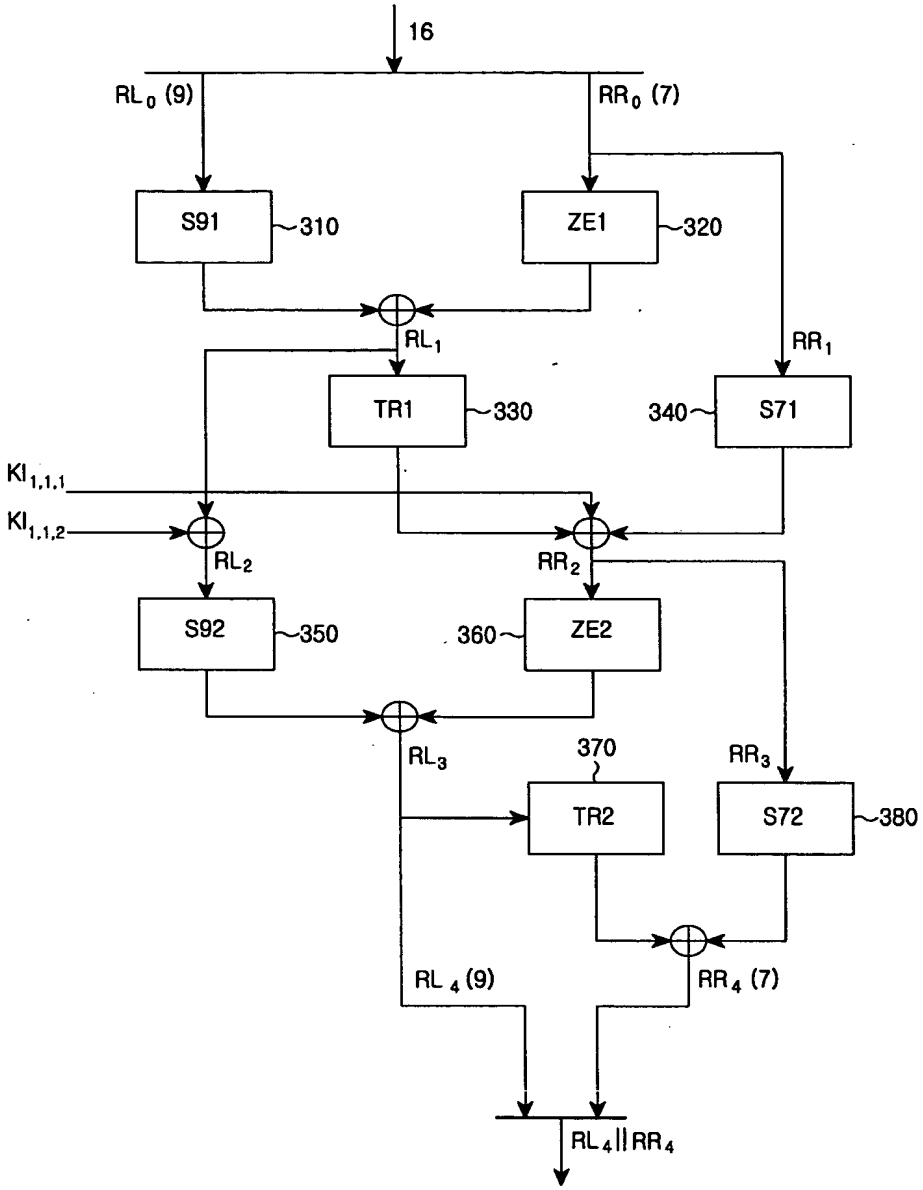
【도 2a】


32
L0 (16)
R0 (16)
R1
KO1,1
L1
KI1,1
FI1,1
D1
10
201
L1D
R1D
단계 1
KO1,2
R2
L2
KI1,2
FI1,2
D2
20
203
L2D
R2D
단계 2
KO1,3
R3
L3
KI1,3
FI1,3
D3
30
205
L3D
R3D
L4 (16)
R4 (16)
단계 3
L4|| R4 (32)

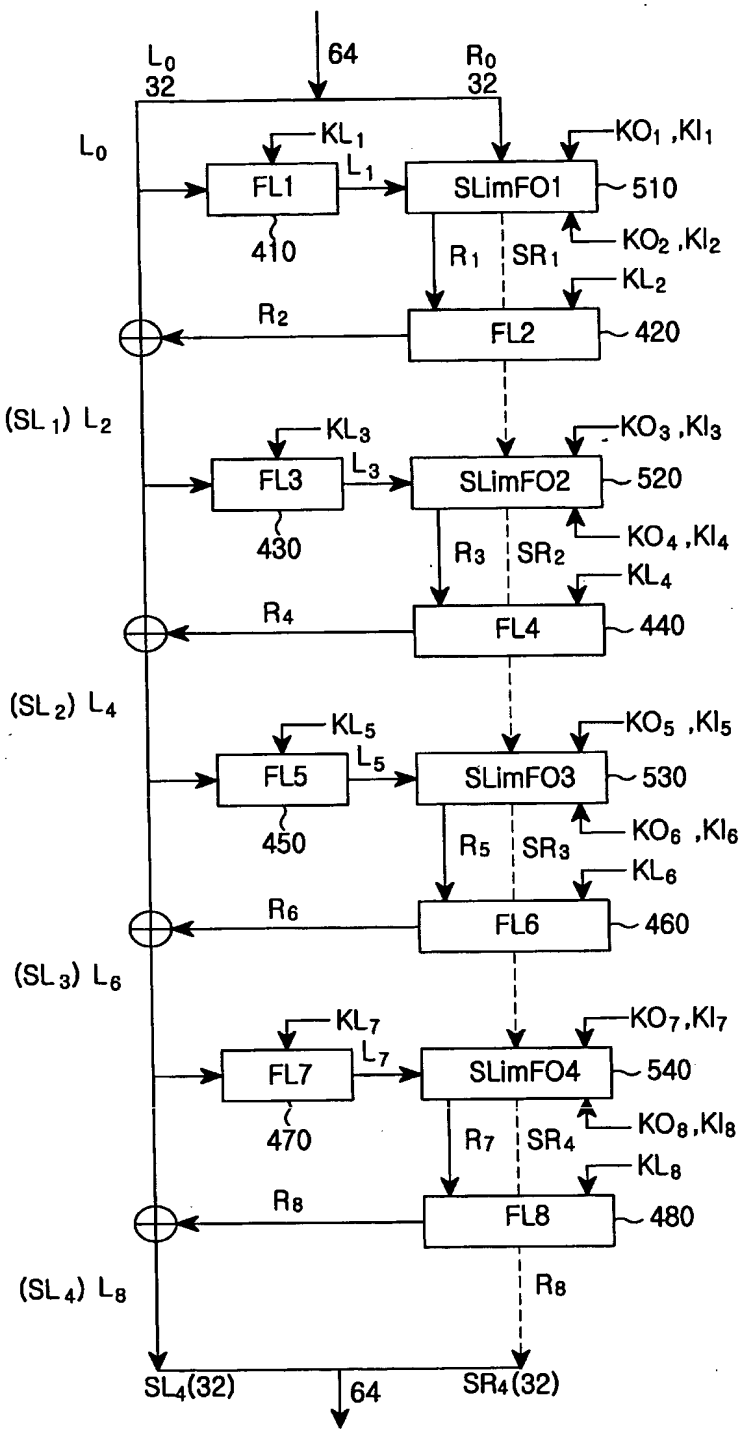
【도 2b】


32
16
16
L0'
R0' (R1')
KO1,1
211
L1'
213
R2'
KO1,2
KI1,1
FI1',1'
D4
40
FI1',2'
KI1,2
L1D'
R1D'
R2D'
L2'
R3'
KO1,3
215
L3'
KI1,3
FI1',3'
D5
50
L3D'
R3D'
R4' (16)
L4' (16)
L4' || R4' (32)

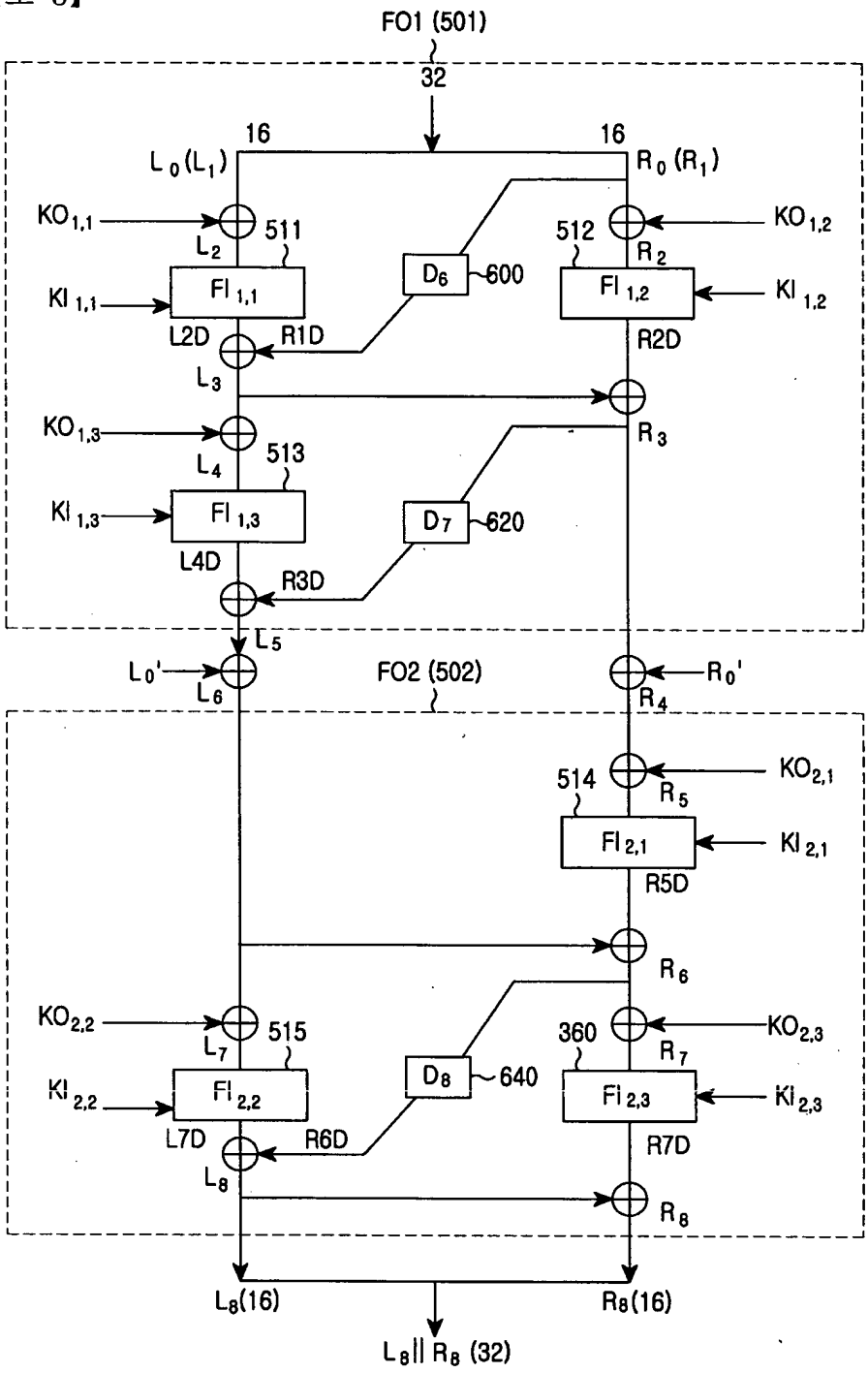
【도 3】


16
RL0 (9)
RR0 (7)
S91
310
ZE1
320
RL1
RR1
TR1
330
340
S71
KI1,1,1
KI1,1,2
RL2
RR2
S92
350
360
ZE2
RL3
370
RR3
TR2
S72
380
RL4 (9)
RR4 (7)
RL4||RR4

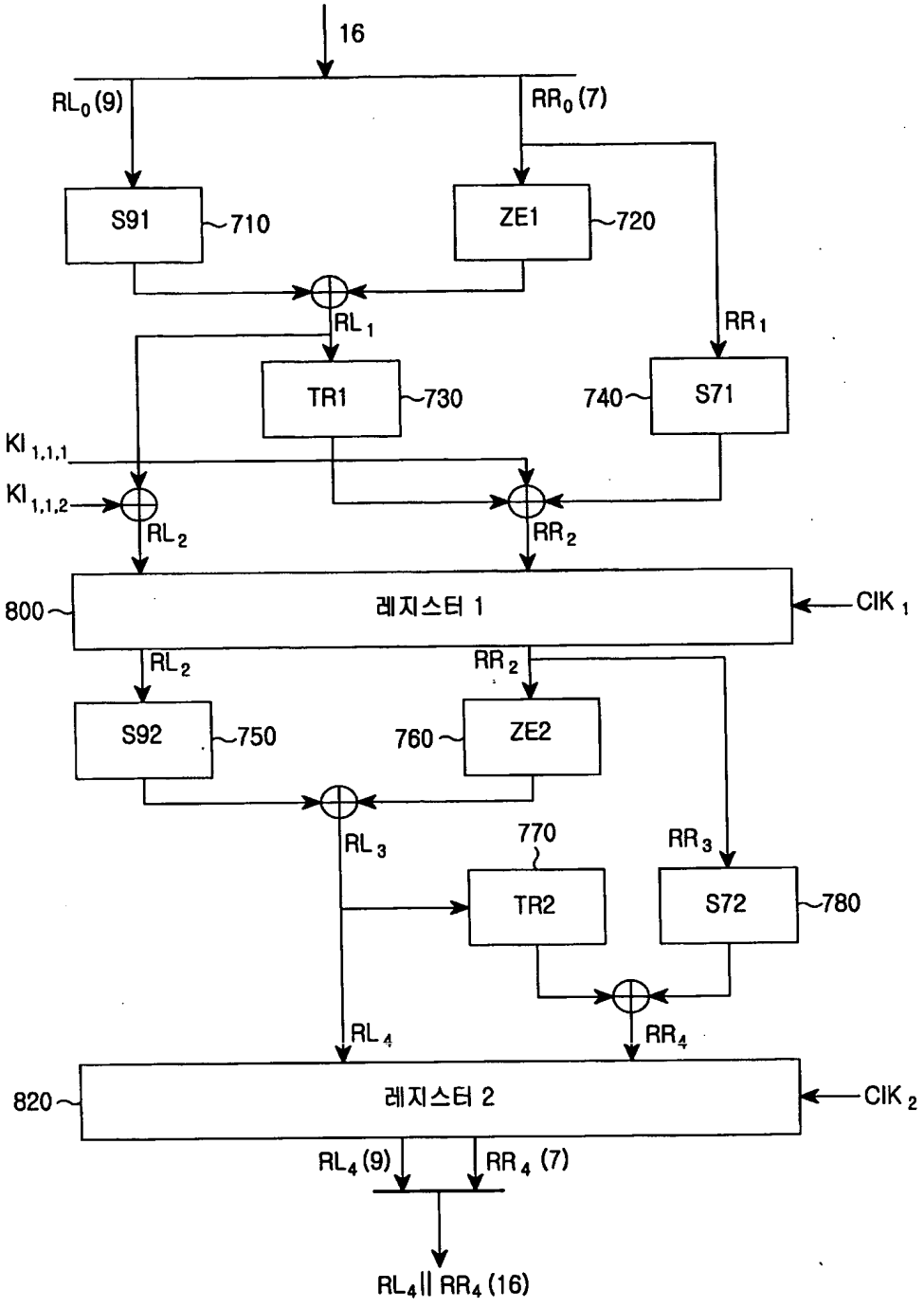
【도 4】


L0
32
64
R0
32
L0
KL1
FL1
L1
SLimFO1
510
KO1 ,KI1
410
R1
SR1
KO2 ,KI2
KL2
R2
FL2
420
(SL1) L2
KL3
FL3
L3
SLimFO2
520
KO3 ,KI3
430
R3
SR2
KO4 ,KI4
KL4
R4
FL4
440
(SL2) L4
KL5
FL5
L5
SLimFO3
530
KO5 ,KI5
450
R5
SR3
KO6 ,KI6
KL6
R6
FL6
460
(SL3) L6
KL7
FL7
L7
SLimFO4
540
KO7,KI7
470
R7
SR4
KO8,KI8
KL8
R8
FL8
480
(SL4) L8
R8
SL4(32)
64
SR4(32)

【도 5】


FO1 (501)
32
16
16
L0(L1)
R0(R1)
KO1,1
511
L2
512
R2
KO1,2
KI1,1
FI1,1
D6
600
FI1,2
KI1,2
L2D
R1D
R2D
L3
KO1,3
R3
L4
513
KI1,3
FI1,3
D7
620
L4D
R3D
L5
L0'
L6
FO2 (502)
R0'
R4
514
KO2,1
R5
FI2,1
KI2,1
R5D
R6
KO2,2
515
L7
360
KO2,3
R7
KI2,2
FI2,2
D8
640
FI2,3
KI2,3
L7D
R6D
R7D
L8
R8
L8(16)
R8(16)
L8||R8 (32)

【도 6】


16
RL0(9)
RR0(7)
S91
710
ZE1
720
RL1
RR1
TR1
730
740
S71
KI1,1,1
KI1,1,2
RL2
RR2
800
레지스터 1
CIK1
RL2
RR2
S92
750
760
ZE2
RL3
770
RR3
TR2
S72
780
RL4
RR4
820
레지스터 2
CIK2
RL4(9)
RR4(7)
RL4||RR4(16)